

# Webserver - Apache

Markus Meier

27. Juni 2011

# Installing Apache

Installation: `apt-get install apache2`

Reachable at: `http://localhost` or `http://192.168.16.<offset>`

The default repository: `/var/www/`

The default webpage: `/var/www/index.html`

# Virtual Hosts

Virtual hosts can be defined in sites

Available sites:

/etc/apache2/sites-available/

Enabled sites:

/etc/apache2/sites-enabled/

To enable or disable a site one can use:

a2ensite <site-name>

a2dissite <site-name>

Afterwards the apache should reload its configuration files.

We want to create the new site

/etc/apache2/available-sites/<uid>

# Virtual Hosts

<VirtualHost \* >

ServerName <uid>

ServerAlias <uid>.course

DocumentRoot /var/www

ServerAdmin <e-mail-address>

ErrorLog /var/log/apache2/error\_<uid>.log

LogLevel warn

<Location>

Options Indexes FollowSymLinks MultiViews

Allow from all

</Location>

</VirtualHost>

# Let Apache do the Virtual Host

In order to be sure, that our new virtual host is used, we disable default and default-ssl:

```
a2dissite default
```

```
a2dissite default-ssl
```

And enable our site:

```
a2ensite <uid>
```

To accept the change apache needs to reload its configuration  
`/etc/init.d/apache2 restart`

# Security - HTTPS

HTTP-protocol: communication between server and webbrowser is not encrypted  $\Rightarrow$  everybody in the internet could simply read the transmissions

HTTPS-protocol: communication between server and webbrowser is encrypted; no additional software is needed  $\Rightarrow$  server and client check the identity of the other one; a key is exchanged between them; this key is used for encryption

HTTPS uses the port 443

# Certificate

To allow the checking of the server identity, the server needs a ssl-certificate.

```
mkdir /etc/apache2/ssl  
cd /etc/apache2/ssl  
openssl req -new -x509 -nodes -out <uid>.course.crt -keyout  
<uid>.course.key
```

Interactive:

Country Name (2 letter code) [AU]:DE

State or Province Name (full name) [Some-State]:Bavaria

Locality Name (eg, city) []:Garching

Organization Name (eg, company) [Internet Widgits Pty  
Ltd]:Rostlab

Organizational Unit Name (eg, section) []:

Common Name (eg, YOUR name) []:<uid>

Email Address []:<your e-mail-address>

# Apache and its Modules - Modules and its Apache

The function of Apache can be extended by installing and/or enabling new modules.

Some modules can be found with a short description by:  
aptitude search apache2-mod-

To enable a module: `a2enmod <module>`

To disable a module: `a2dismod <module>`

In both cases apache must reload its config-files.



# SSL and Apache

We need to change the configuration of the virtual host, such that it uses ssl.

```
<VirtualHost [youripaddress]:443>  
...  
SSLEngine On  
SSLCipherSuite HIGH:MEDIUM  
SSLCertificateFile /etc/apache2/ssl/<uid>.course.crt  
SSLCertificateKeyFile /etc/apache2/ssl/<uid>.course.key  
...  
</VirtualHost>
```

Afterwards we have to enable the already available module ssl:  
a2enmod ssl

And restart apache:  
/etc/init.d/apache2 restart

# Apache and PHP

Installation: `apt-get install libapache2-mod-php5`

Aptitude will automatically remove `apache2-mpm-worker` which is not applicable to php, enable the newly installed `php5` module and restarts the server.

In order to create an info php-script accessible by the server we just have to create the file `/var/www/test.php`:

```
<?php phpinfo(); ?>
```

Accessible at `http://<uid>/test.php`

## More security - ip-restriction

We can specify allowed ip-addresses:

Allow from 192.168.16.5 192.168.16.15

Or a whole subnet:

Allow from 192.168.16

Or deny ip-addresses:

Deny from 192.168.16.2

With order we can say which comes first: restriction or allow

Order Allow,Deny

## More security - defined user/password

Adding a user/password pair to a new htpasswd

```
htpasswd -c /etc/apache2/htpasswd MyUserName
```

Adding/updating a user/password pair to/of an existing htpasswd

```
htpasswd /etc/apache2/htpasswd MyUserName
```

Deleting a user with its password from htpasswd

```
htpasswd -D /etc/apache2/htpasswd MyUserName
```

AuthType Basic

AuthName "Internal area - Authorized users only"

AuthUserFile /etc/apache2/htpasswd

Require valid-user

# Apache and LDAP - phpldapadmin

- ▶ apt-get install phpldapadmin
- ▶ Aptitude configures it automatically for our purposes
- ▶ Add /etc/phpldapadmin/apache.conf to /etc/apache2/apache.conf
- ▶ /etc/init.d/apache2 restart
- ▶ Reachable at `http://<uid>/phpldapadmin`

Note: I had to adjust some ownerships, because of some errors in /var/log/apache2/<uid>.course... but before the golden truth hit me, I did some strange things

# Apache and LDAP - ldap-account-manager

- ▶ installing ldap-account-manager:  
apt-get install ldap-account-manager
- ▶ Aptitude does most of the configuration automatically for our purposes.
- ▶ Add /etc/ldap-account-manager/apache.conf to /etc/apache2/apache.conf
- ▶ /etc/init.d/apache2 restart
- ▶ LAM reachable at http://<uid>/lam
- ▶ Change the lam-master-password: http://<uid>/lam > 'LAM configuration' > 'Edit general settings' > default password 'lam' > change the master-password

## Apache and LDAP - ldap-account-manager

Set password and LDAP-server: 'LAM configuration' > 'Edit server profiles' > default password 'lam' > 'General Settings' >

- ▶ In the section Server Settings:
  - ▶ Change Tree suffix to 'dc=course'
- ▶ In the section Security Settings:
  - ▶ Change List of valid users to 'cn=admin,dc=course'
  - ▶ Choose a new password

> save

Set used nodes: http://<uid>/lam > 'LAM configuration' > 'Edit server profiles' > use your new personal password > 'General Settings' >

- ▶ In the section 'Account types':
  - ▶ Delete alle accounts despite Users and Groups
  - ▶ For Users change the LDAP suffix to: ou=people,dc=course
  - ▶ For Groups change the LDAP suffix to: ou=group,dc=course

> save